Remarks

Reconsideration of this application as amended is respectfully requested.

Claims 20-46 stand rejected under 35 U.S.C. §112, second paragraph.

Claims 20-22, 25-28, 31-32, 35-37, 40-41, and 44-46 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,999,732 of *Bak et al.* ("*Bak*") and U.S. Patent No. 5,848,423 of *Ebrahim et al.* ("*Ebrahim*").

Claims 23-24, 33-34, and 42-43 stand rejected under 35 U.S.C. §103(a) as being unpatentable over *Bak* and *Ebrahim* and U.S. Patent No. 5,787,431 of *Shaughnessy* ("*Shaughnessy*").

Claims 29-30, and 38-39 stand rejected under 35 U.S.C. §103(a) as being unpatentable over *Bak* and *Ebrahim* and U.S. Patent No. 6,295,643 of *Brown et al.* ("*Brown*").

The Examiner has rejected claims 20-46 under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The Examiner has stated that the term "predefined definition format" of each of the claims 20, 31, and 40 lacks an antecedent basis. In response, applicant has amended claims 20, 31, and 40 to consistently recite "predefined class definition format" which has the proper antecedent basis.

The Examiner has rejected claims 20-22, 25-28, 31-32, 35-37, 40-41, and 44-46 under 35 U.S.C. §103(a) as being unpatentable over *Bak* and *Ebrahim*. Applicant respectfully submits, however, that amended claim 20 in not obvious in view of *Bak* and *Ebrahim* because *Bak* and *Ebrahim* do not teach or suggest a class loader that enables a virtual machine to obtain a set of classes via a network as needed while executing an application

program as claimed in amended claim 20.

The Examiner has stated that

> Bak discloses ... class loader that enables the virtual machine to obtain a set of classes via a network as needed while executing an application program [col 6, lines 1-6].

(Page 3, Office Action, 3/14/03). The section of *Bak* cited by the Examiner is as follows

> A dynamic class loader and verifier 211 loads Java class files 203 and standard built-in Java classes 205 via operating system 209 into a memory 213. Additionally, the dynamic class loader and verifier may verify the correctness of the bytecodes in the Java class files, reporting any errors that are detected.

(*Bak*, col. 6, lines 1-6). Applicant submits that there is nothing in this statement to even suggest that *Bak* loads class files via a network as needed by an application program as claimed in amended claim 20. Fig. 4 of *Bak* shows the Java class files 203 as an input to a Java runtime system 201 without any indication of network access.

Applicant further submits that *Bak* and *Ebrahim* do not teach or suggest a memory manager that selects and purges the arrays and references from a class structure so as to minimize class loading activities on a network as claimed in amended claim 20. The Examiner has stated that

> *Ebrahim* discloses the memory manager [garbage collector] that selects and purges the arrays and references of the classes from the class structures so as to minimize an amount of the memory consumed by the class structure...

(Page 3, Office Action, 3/14/03).

Applicant respectfully submits that *Ebrahim* teaches recovering memory space in a heap 116 (*Ebrahim*, col. 2, lines 36-38) that stores instances of object classes stored in a class repository 150 (*Ebrahim*, col. 3, lines 40-42) rather than purging arrays and references from a class structure as claimed in amended claim 20.

Furthermore, given that *Ebrahim* does not teach obtaining classes via a network as claimed in amended claim 20, it follows that *Ebrahim* cannot teach selecting an purging the arrays and references from a class structure so as to minimize class loading activities on a network as claimed in amended claim 20.

Given that claims 21-30 depend from amended claim 20, it follows that claims 21-30 are not obvious in view of *Bak* and *Ebrahim*.

Applicant also submits that amended claims 31 and 40 are not obvious in view of *Bak* and *Ebrahim*. Amended claims 31 and 40 include limitations similar to the limitations in amended claim 20 including obtaining a set of classes via a network as needed while executing an application program and purging the arrays and references from a class structure so as to minimize class loading activities on a network. Therefore, the remarks stated above with respect to amended claim 20 also apply to amended claims 31 and 40.

Given that claims 32-39 and 41-46 depend from amended claims 31 and 40, it is submitted that claims 32-39 and 41-46 are not obvious in view of *Bak* and *Ebrahim*.

It is also submitted that claims 23-24, 33-34, and 42-43 are not obvious in view of *Bak* and *Ebrahim* and *Shaughnessy*. Claims 23-24, 33-34, and 42-43 depend from amended claims 20, 31, and 40 and applicant has shown that amended claims 20, 31, and 40 are not obvious in view of *Bak* and *Ebrahim* because *Bak* and *Ebrahim* do not teach or suggest obtaining a set of classes via a network as needed while executing an application program and purging the arrays and references from a class structure so as to minimize class loading activities on a network. *Shaughnessy* does not teach or suggest obtaining a set of classes via a network as needed while executing an application program and purging the arrays and references

from a class structure so as to minimize class loading activities on a network. Instead, *Shaughnessy* discloses a reference cache that holds pointers to strings (*Shaughnessy*, col. 4, lines 40-41) that are used for naming database columns (*Shaughnessy*, col. 4, lines 35-39).

It is further submitted that claims 29-30, and 38-39 are not obvious in view of *Bak* and *Ebrahim* and *Brown*. Claims 29-30, and 38-39 depend from amended claims 20 and 31 and applicant has shown that amended claims 20 and 31 are not obvious in view of *Bak* and *Ebrahim* because *Bak* and *Ebrahim* do not teach or suggest obtaining a set of classes via a network as needed while executing an application program and purging the arrays and references from a class structure so as to minimize class loading activities on a network. *Brown* does not teach or suggest obtaining a set of classes via a network as needed while executing an application program and purging the arrays and references from a class structure so as to minimize class loading activities on a network. Instead, *Brown* discloses a system for generating FCCF files that contain information recorded while executing classes of an application program. (*Brown*, col. 8, lines 20-22).

It is respectfully submitted that in view of the amendments and arguments set forth above, the applicable rejections have been overcome.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 08-2025 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: 6-16-03 By: _____

Paul H. Horstmann
Reg. No.: 36,167